# A Resource Scheduling Model for Cloud Computing Data centers

Mohamed Abu Sharkh, Abdelkader Ouda, and Abdallah Shami

Department of Electrical and Computer Engineering, Western University

London, Canada

{mabusha, ashami2, aouda}@uwo.ca

***Abstract*—Cloud computing is an increasingly popular computing paradigm, now proving a necessity for utility computing services. Each provider offers a unique service portfolio with a range of resources. Resource provisioning for cloud services in a comprehensive way is of crucial importance to any resource allocation model. Any model should consider both computational resources and network(data) resources to accurately represent and serve practical needs. We propose a new model to tackle the resource allocation problem for a group of cloud user requests. This includes provisioning for both data center computational resources and network resources. The model is implemented with the objective of minimizing the average tardiness of connection requests. Four combined scheduling algorithms are introduced and used to schedule virtual machines on data center servers and then schedule connection requests on the network paths available. Of the four methods, the method combining Resource Based Distribution technique and Duration Priority technique have shown the best performance getting the minimum tardiness while complying with the problem constraints.**

***Keywords: Cloud Computing, SaaS, Virtual networks, data centers, SDN controller.***

## I. Introduction

Cloud computing is an increasingly popular computing paradigm, now proving a necessity for utility computing services. Several providers have cloud computing solutions available, where a pool of virtualized and dynamically-scalable computing power, storage, platforms, and services are delivered on demand to clients over the Internet in a pay as you go manner. This is implemented using large data centers where thousands of servers reside. Each provider offers a unique service portfolio with a range of options that include virtual machines (VMs) instance configuration, nature of network services, degree of control over the rented machine, supporting software/hardware security services, additional storage, etc.

The diversity in instance types, in multiple data centers in different locations, makes resource management even more of a complicated matter. The resource management of the data centers servers while scheduling and serving tens of thousands of client requests on VMs residing on data center servers, is a critical success factor. In fact, it is a main revenue source to the service provider as excess resources translate directly to revenue.

### A. Related Work

*1- Efforts with a focus on data center processing Resources:* Multiple models were proposed previously, where resources are scheduled based on user requests. In [1], a queuing model is proposed where a client requests virtual machines for a predefined duration. VM pre-emption, one and multiple job queues are considered. Network Resources are not considered at all. Jobs are assumed not to communicate with each other or transmit or receive data. No preference is required as to where the VMs are to be scheduled. In [2], an algorithm is proposed to optimally distribute VMs in order to minimize the distance between user VMs in a data center grid. The only network constraint used is the Euclidean distance between data centers. No specific connection requests or user differentiation is used. In the same paper, an algorithm is proposed to schedule VMs on racks, blades and processors within one data center to minimize communication cost. In [3], three consecutive phase queues are used to schedule prioritized job categories. No network topology is used. Rather, only the monetary cost of transmitting data is considered for network requests.

*2- Efforts with a focus on Data center Network Resources:* In [4], the authors tackle the problem where a client may have multiple jobs being processed at the same time but not necessarily on the same server. Requests are abstracted as a virtual network (VN) where VMs represent nodes and paths between two nodes represent VN links (edges). The problem is treated as an optimization problem of provisioning a virtual network with the objective of revenue maximization. The time factor is not provisioned for since no reservation start time or duration is introduced. The scenario where a user requests more connectivity for an already reserved VM is not considered. In [5], authors tackle the problem of proposing the best virtual network with IP over wavelength division multiplexing (WDM) network. Only network demands are considered with three different network profile demands . Constraints are based on propagation delay, flow conversion constant and capacity. The user demand for processing resources is not considered. The authors target minimizing the network power consumption by minimizing the number of network components that are switched on.

978-1-4673-2480-9/13/$31.00 ©2013 IEEE


Authorized licensed use limited to: Mukesh Patel School of Technology & Engineering. Downloaded on July 11,2023 at 05:39:03 UTC from IEEE Xplore. Restrictions apply.

### B. Motivation

Provisioning for cloud services in a comprehensive way is of crucial importance to any resource allocation model. Any model should consider both computational resources and network resources to accurately represent practical needs. First, excluding the processing (computational resources), while designing the resource allocation model, deprives the model of the main cloud service. Cloud data centers are built first and foremost as ways to outsource computational tasks. Any model that optimizes data center resources should include answers to questions like : how are VMS allocated? how are processing resources modeled? what is the resource portfolio that is being promoted to clients? How are the data center resources distributed physically? On the other hand, Network services are the backbone of the cloud computation services. As clients ask for tasks to be processed in the data center, they need networking service with adequate QoS standards to send and receive the application data. It is no use to clients if their application is producing the results needed in the required time if these results can not be delivered to the client base through a stable network connection on time. In [6], data transfer bottlenecks are stated as one of the main obstacles cloud client base growth is facing. Authors show that when moving large amounts of data in a distributed data center environment, the network service performance will be the critical point for the whole process. In the example mentioned, the authors reached the conclusion that for some configurations the data transmission tardiness would cause the client to choose options like sending data disks with a courier (FedEx, for example). This is not a a not-so-frequent problem as it is a common practice to distribute client data over multiple geographically distant data centers with Internet as the communication medium more often than not. As a result, cloud clients will see network service specification as a main factor in their decision to move to the cloud (or to choose which cloud provider). Factors like bandwidth options, port speed, number of IP addresses, load balancing options, availability of VPN access among others should be considered by any comprehensive model.

The proposed model in this paper tackles the resource allocation challenges faced when provisioning computational resources (CPU, memory, storage) and network resources. Advance or immediate reservation requests for resources are sent by clients. A central controller manages these requests with the objective of minimizing average tardiness and request blocking. The solution aims at solving the provider's cost challenges and the cloud applications performance issues. To the best our knowledge, this paper is the first to address the resource allocation problem in cloud computing data centers in this form. The proposed model schedules network connection requests for the network between data centers and client nodes. A future step will be tackling internal data center networks scheduling.

This paper is organized as follows: In Section II, the proposed model is discussed in detail. After that, the proposed heuristic techniques used during the experiments are introduced in Section III. Simulation environment and results are shown in Sections IV and V. Finally, future steps are described in the conclusion.

## II. Used Model

This paper considers the challenges every provider is facing when planning and operating the data centers. The main goal of this work is to study and analyze the available resource management systems and provide a system that maximizes the data center resource utilization in cloud computing environments and minimizes the service provider cost while fulfilling all the clients' requests at the same time.

In this work, resource allocation is coupled with the concepts of software defined networking (SDN). SDN is a networking paradigm in which the forwarding behavior of a network element is determined by a software control plane decoupled from the data plane. SDN leads to many benefits such as increasing network and service customizability, supporting improved operations and increased performance. The software control plane can be implemented using a central network controller. We are proposing that the central controller will handle the task of resource allocation in the network between the data centers. This can be done by directing all the client requests to the SDN controller. As discussed later in detail, the SDN controller will execute the resource allocation algorithms. It will then send the allocation commands across the network.

The model consists of a network of data centers nodes (public clouds) and client nodes (private clouds) . These nodes are located in different cities or geographic points. They are connected using a network of bidirectional links. Every link in this network is divided into a number of equal lines (flows). We assume that this granularity factor of the links can be controlled. We also assume each data center contains a number of servers connected through Ethernet connections. Each server will have a fixed amount of memory, computing units and storage space.

As an initial step, when clients require cloud hosting, they send requests to reserve a number of VMs. These VMs can be all of the same type or of different types. Each cloud provider offers multiple types of VMs for their clients to choose from. These types vary in the specification of each computing resource (memory, CPU units, storage, etc). We will use the three types shown in Table 2 in our experiment.

The SDN controller allocates each of the requested VMs on a server in one of the data centers. Also, the client sends a number requests to reserve a connection. There are two types of connection requests:

1- A request to connect a VM to another VM. Here, Both VMs were previously allocated space on a server in one of the data centers (public clouds).

2- A request to connect a VM to a client node. Here, the VM located in a data center node connects to the client headquarters or private cloud.

For every request, the client defines the source, destination, start time and duration of the connection. Thus, this problem



Authorized licensed use limited to: Mukesh Patel School of Technology & Engineering. Downloaded on July 11,2023 at 05:39:03 UTC from IEEE Xplore. Restrictions apply.

TABLE I
VM CONFIGURATION FOR THE 3 INSTANCE (VM) TYPES USED IN THE EXPERIMENT AS OFFERED BY AMAZON EC2 [7].

| Instance Type | Standard Extra large (SXL) | High Memory Extra Large (MXL) | High CPU Extra Large (CXL) |
|---|---|---|---|
| Memory | 15 GB | 17 GB | 7 GB |
| CPU (EC2 units) | 8 | 6.5 | 20 |
| Storage | 1690 GB | 490 GB | 1690 GB |

falls under the advance reservation category of problems.

The SDN controller keeps the data tables of the avilable network paths, available server resources and connection expiration times in order to handle newly arriving requests. The SDN allocates the requested VMs on servers according to the method or policy used. It updates the resource availability tables accordingly. After that, the SDN controller schedules and routes connection requests to satisfy the client requirements. Network path availability tables are also updated. As an initial objective, the SDN controller aims at minimizing the average tardiness of all the advance reservation connection requests. Also , a second objective is minimizing the blocked requests. This objective is to be reached regardless of what path is used.

## III. HEURISTIC TECHNIQUES FOR MINIMIZING TARDINESS

The allocation process is divided into two subproblems:

1- Allocation of VMs on data center servers. Here, all the VM reservation requests are served based on server resource availability before any connection request is served.

2- Scheduling of connection requests on the available network paths. This happens after all VMs have been allocated resources and started operation on the servers.

For the first subproblem, three heuristic techniques were evaluated. For the second subproblem, two heuristic techniques were tested. For a complete experiment, one heuristic for each subproblem is used. These heuristics are divided as follows :

### A. VM reservation heuristic techniques

a) Equal Time Distribution Technique:
In this heuristic, $TM_i$ is the total time reserved by connection requests from the virtual machine $VM_i$ (sum of the connection durations). Next, the share of one server is calculated by dividing the total time units all the VMs have requested by the number of servers. Then, for each server, VMs are allocated computation resources on the server one by one. When the server share is fulfilled, the next VM is allocated resources on the following server and the previous steps are repeated. The algorithm is described in pseudo code in Figure 1.

b) Node Distance Technique:
First, the average distance between each two nodes is calculated. The two nodes furthest from each other (with maximum distance) are chosen. Then , the maximum number of VMs is allocated on the servers of these two nodes. Next, the remaining nodes are evaluated, the node with maximum average distance to the previous two nodes is chosen. The same process is repeated until all the VMS are scheduled. The algorithm is described in pseudo code in Figures 2 and 3.

c) Resource Based Distribution Technique:
In this heuristic, the choice of the server is based on the type of VM requested. As shown in Table 1, three types of VMs are used in the experiment: i) High Memory Extra Large (MXL) has high memory configuration; ii) High CPU Extra Large (CXL) has a high computing power; iii)Standard Extra large (SXL) is more suited to applications that need a lot of storage space. Depending on the type of VM requested by the client, the heuristic picks the server with the highest amount of the available resources. The VM then is allocated reosurces on that server. This causes the distribution to be more balanced.

### B. Connection reservation heuristic techniques

a) Duration Priority Technique:
In this heuristic, connections with the shortest duration are given the priority. First, connection requests are sorted based on the requested duration. The following step is to pick the connection with the shortest duration and schedule it on the on the shortest path available. This step is repeated until all connection requests are served. The algorithm is described in pseudo code in Figure 4.

b) Greedy Algorithm:
In this heuristic, scheduling is based on the connection requested start time (RST). Connection requests with earlier RST are scheduled on the first path available regardless of the path length. The algorithm is described in pseudo code in Figure 5.

## IV. SIMULATION

The problem is simulated using a dicrete event based simulation program and solved on a more practical scale using the heuristic search techniques discussed in the previous section. The network used for the experiment is NSF network. It consists of 14 nodes of which 3 are data center nodes and the rest are considered client nodes. Nodes are connected using a high speed network with link granularity chosen as 3 lines (flows) per link. Fixed alternate routing method is used with 3 paths available between a node and any other node. Server configuration and request data parameters are detailed in Table 2. Pre-emption of connection requests is not allowed in the experiment described in this paper.

## V. RESULTS

As explained in the previous sections, every experiment inlcudes two subproblems and hence two heuristics are needed: one to schedule VMs on servers and the other to schedule connection requests. The different simulation scenarios and combined heuristics used for the two subproblems are as follows:

1- ED-GA : Equal Time Distribution technique and Greedy algorithm.



Authorized licensed use limited to: Mukesh Patel School of Technology & Engineering. Downloaded on July 11,2023 at 05:39:03 UTC from IEEE Xplore. Restrictions apply.

**Input:** $Virtual\ machine\ set\ VM,\ Server\ set\ Q,\ connection\ request\ set\ R$
**Output:** $Allocation\ of\ VMs\ on\ servers$
**for** $TM_i \in TM$ **do**
  $TM_i = 0$
**end for**
**for** $VM_i \in VM$ **do**
  **for** $R_j \in R$ **do**
    **if** $R_j.source = VM_i$ **or** $R_j.dest = VM_i$ **then**
      $TM_i = TM_i + R_j.duration$
    **end if**
  **end for**
**end for**
$TM_{total} = \sum_i TM_i$
$ServerShare = TM_{total}\ /|Q|$
$i = 0$
**for** $S_j \in Q$ **do**
  $ThisServerShare = ServerShare$
  **while** $S_j\ isNotFull$ **and** $ThisServerShare > TM_i$ **do**
    $Schedule\ VM_i on S_j$
    $i = i + 1$
    $ThisServerShare = ThisServerShare - TM_i$
  **end while**
**end for**

Fig. 1. Equal time distribution heuristic technique

TABLE II
EXPERIMENT PARAMETER CONFIGURATION.

| Parameter | Value |
|---|---|
| Total number of servers | 132 |
| Servers/ data center | 44 |
| VM reservation requests | 200 |
| Connection requests | 10000 |
| RST distribution | Poisson with Lambda = 10 |
| Connection duration distribution | Normal with mean = 200 time units |
| Source and destination distribution | Uniform |
| Allowed tardiness per request | values ranging from 1 to 500 time units |
| Total experiment time | 70,000 time units |

2- RB-DP : Resource Based Distribution technique and Duration Priority technique.

3-ED-DP : Equal Time Distribution technique and Duration Priority technique.

4-ND-DP: Node Distance technique and Greedy algorithm.

In Figures 7 and 8, the results for blocking percentage are shown for the four methods. Figure 7 shows a comparison between the blocked requests percentage produced when using each one of the four methods where the allowed tardiness is very small (1 time unit). Figure 8 shows the same comparison when the allowed tardiness per request is large (30000 time units ). It can be seen from both figures that ED-DP and RB-DP methods have shown a clear advantage. This indicates a clear advantage of using DP over GA when scheduling connection requests in tight or real time conditions. As it shows in Figure 8, RB-DP has shown a decent advantage over

**Input:** $Virtual\ machine\ set\ VM,\ Server\ set\ Q,\ Node\ set\ N\ Path\ set\ P\ where\ P_{ijk}\ is\ path\ k\ between\ nodes\ i\ and\ j\ is\ NP\ Number\ of\ paths\ between\ node\ i\ and\ node\ j$
**Output:** $Allocation\ of\ VMs\ on\ servers$
**for** $N_i \in N$ **do**
  **for** $N_j \in N$ **do**
    $A[i][j] = \sum_k^N PP_{ijk}.Length/NP$
  **end for**
**end for**
$Pick\ 2\ nodes\ x,\ y\ with\ max\ A[x][y]$
$U = \{x, y\}$
$RequestedVMs = |VM|$
$fillNode(x, RequestedVMs)$
$fillNode(y, RequestedVMs)$
**while** $U \neq N$ **and** $RequestedVMs > 0$ **do**
  $maxDist = 0$
  **for** $N_i \in N$**and**$N_i \notin U$ **do**
    $avgDet = 0$
    **for** $B_j \in U$ **do**
      $avgDist = avgDist + A[B_j][N_i]$
    **end for**
    **if** $avgDist > maxDist$ **then**
      $maxDist = avgDist$
      $NextNode = N_i$
    **end if**
  **end for**
  $fillNode(NextNode, requested_VMs)$
  $U = U \cup NextNode$
**end while**

Fig. 2. Node distance heuristic technique

**Function :fillNode**
**Input:** $Virtual\ machine\ set\ VM,\ Node\ x,\ RequestedVMs\ and\ server\ set\ Q$
**Output:** $servers\ in\ node\ x\ filled\ with\ max\ VMs\ possible$
$i = 0$
**for** $S_i \in\ Q\ and\ S_i\ residing\ in\ Node\ x$ **do**
  **while** $S_i\ isNotFull$ **and** $i < RequestedVMs$ **do**
    $Schedule\ VM_i on S_j$
    $i = i + 1$
  **end while**
**end for**

Fig. 3. Function :fillNode



Authorized licensed use limited to: Mukesh Patel School of Technology & Engineering. Downloaded on July 11,2023 at 05:39:03 UTC from IEEE Xplore. Restrictions apply.

**Input:** *Path set P where $P_{xyk}$ is path k between nodes x and y,* Server *et Q, and connection request et R*
**Output:** *Scheduling of network connection requests on network paths*
*Sort R in descending order based on $R_i.duration$*
**for** $R_i \in R$ **do**
  **for** $t = R_i.RST$ *to* $MaxTimeUnits$ **do**
    *Pick shortest path $P_{xyk}$ where $R_i.source = x$ and $R_i.destination = y$*
    **if** $P_{xyk}$ *isAvailable*$(t, R_i.duration)$ **then**
      *Schedule $R_i$ on $P_{xyk}$ at time unit t*
      *Move to next request*
    **end if**
  **end for**
**end for**

Fig. 4. Duration priority heuristic technique

**Input:** *Path set P where $P_{xyk}$ is path k between nodes x and y is Server set Q, connection request set R, NP Number of paths between node i and node j*
**Output:** *Scheduling of network connection requests on network paths*
*Sort R in descending order based on $R_i.RST$ (requested start time)*
**for** $R_i \in R$ **do**
  **for** $t = R_i.RST$ *to* $MaxTimeUnits$ **do**
    $x = R_i.source$
    $y = R_i.destination$
    **for** $k = 0$ *to* $NP$ **do**
      **if** $P_{xyk}$*isAvailable*$(t, R_i.duration)$ **then**
        *Schedule $R_i$ on $P_{xyk}$ at time unit t*
        *Move to next request*
      **end if**
    **end for**
  **end for**
**end for**

Fig. 5. Greedy heuristic technique

ED-DP in terms of blocking percentage.

As for the average tardiness per request, the measurements are shown in Figure 9. The figure shows a comparison between the average tardiness per request produced when using each one of the four methods, where the allowed tardiness is small (25 time unit). Once more, ED-DP and RB-DP methods have shown a clear advantage. Also, it is noticed from the figure the ED-DP produces slightly better results (less average tardiness) than RB-DP. Therefore, using RB-DP method is more suitable to scenarios where there is an emphasis on serving the largest number of requests. On the other hand, using ED-DP is more suitable to scenarios where the individual request performance

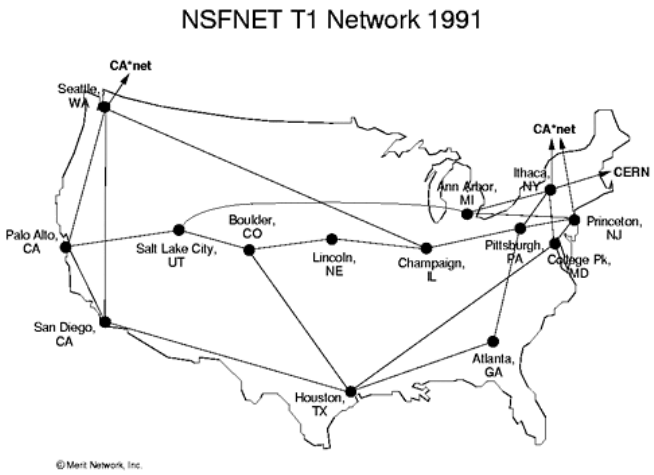


Fig. 6. Shows the NSF network of 14 nodes.

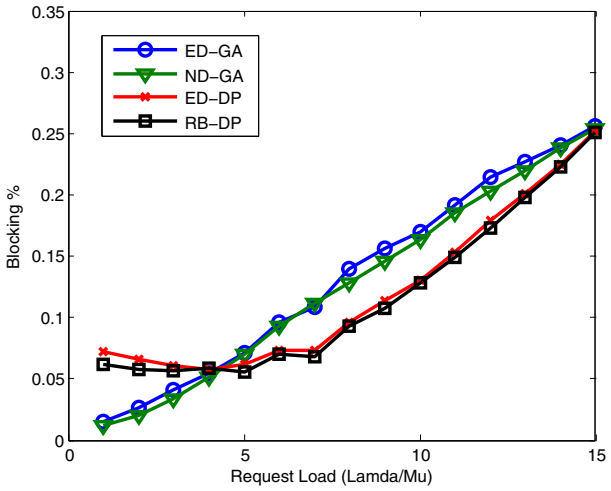


Fig. 7. Request Blocking results for scheduling methods (allowed tardiness/request =1 time units

or service level is prioritized over serving more requests.

## VI. CONCLUSION

A new model that tackles the resource allocation problem for a group of cloud user requests was presented. This model uses virtual network concepts and combines provisioning network resources with data center computational resources. The model is implemented with the objective of minimizing the average tardiness of connection requests and minimizing the connection blocking percentage. Five heuristic techniques were evaluated in the simulation. Combining those techniques, four methods were introduced and used to schedule virtual machines on data center servers and schedule connection requests on the network paths available. Simulation results and comparison of these methods' performance were discussed. Of the four methods, the method combining Resource Based Distribution technique and Duration Priority technique (RB-DP)



Authorized licensed use limited to: Mukesh Patel School of Technology & Engineering. Downloaded on July 11,2023 at 05:39:03 UTC from IEEE Xplore. Restrictions apply.

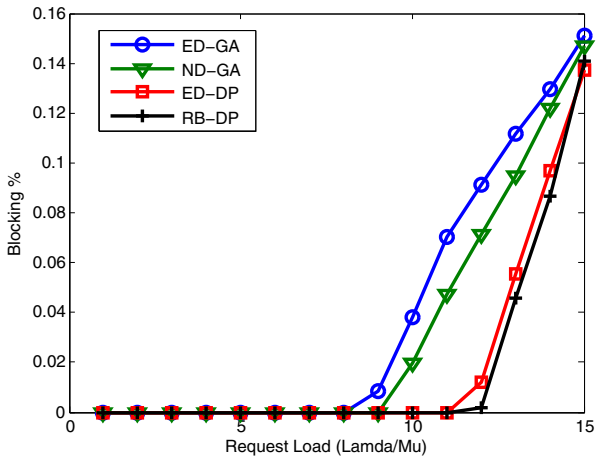


Fig. 8. Request Blocking results for scheduling methods (allowed tardiness/request =30000 time units

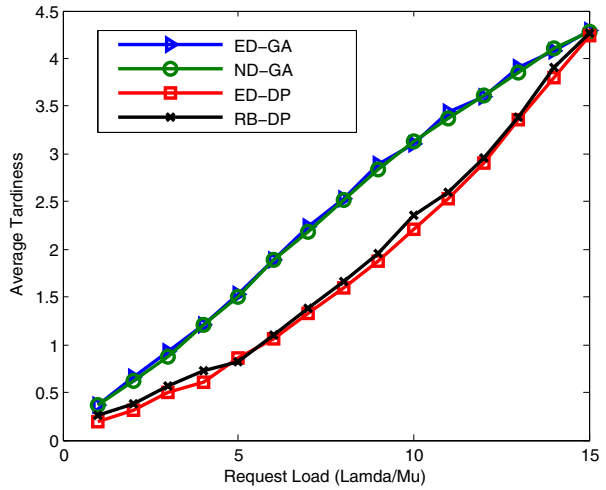


Fig. 9. Request averAge tardiness results for scheduling methods (allowed tardiness/request =25 time units

have shown the best performance in terms of request blocking percentage while complying with the problem constraints. The method combining Equal Time Distribution technique and Duration Priority technique (RB-DP) have shown the best performance in terms of avergae tardiness per request while complying with the problem constraints.In the future work, In addition to enhancing the performance of this system, we would like to investigate issues like using multiple SDN controllers to control the same network. Moreover, we aim at testing features like pre-emption or fixing the allocated intervals to requests. These features can be seen as a possible performance push.We would also like to enhance our model to include another objective which is minimizing power consumption. After measuring the power consumption performance, the following step would be to combine heuristics to solve the bi-objective problem of minimizing power consumption and tardiness at the same time.

## REFERENCES


[1] S. Maguluri, R. Srikant, and L. Ying, *" Stochastic Models of Load Balancing and Schedulingin Cloud Computing Clusters,"* IEEE INFOCOM 2012 Proceedings. pp.702-710, 25-30 Mar,2012.

[2] M. Alicherry, and T.V. Lakshman. *"Network aware resource allocation in distributed clouds,"* IEEE INFOCOM 2012 Proceedings. pp.963-971, 25-30 Mar,2012.

[3] X. Nan, Y. He, and L. Guan, *" Optimal resource allocation for multimedia cloud in priority service scheme, "* IEEE International Symposium on Circuits and Systems (ISCAS),2012, pp. 1111 - 1114

[4] G. Sun, V. Anand, H. Yu, D. Liao, and L.M Li, *" Optimal Provisioning for Elastic Service Oriented Virtual Network Request in Cloud Computing,"* IEEE Globecom 2012.

[5] B. Kantarci, and H.T. Mouftah, *"Scheduling Advance reservation requests for wavelength division multiplexed networks with static traffic demands, "* IEEE Symposium on Computers and Communications (ISCC), 2012. pp. 806-811, 1-4 Jul 2012.

[6] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, *"Above the Clouds: A Berkeley View of Cloud Computing"*. Tech. Rep. UCB/EECS-2009-28, EECS Department, U.C. Berkeley, Feb 2009.

[7] *"Amazon Elastic Compute Cloud (Amazon EC2)"* , available online from : http://aws.amazon.com/ec2/.

[8] T.D.Wallas, a, Shami, and C.Assi, *"Scheduling Advance reservation requests for wavelength division multiplexed networks with static traffic demands,"* IET communications, 2008, Vol. 2, No.8. pp. 1023-1033.

[9] I. Foster, Y. Zhao, I. Raicu, and S. Lu, *"Cloud Computing and Grid Computing 360-Degree Compared"*. Grid Computing Environments Workshop, 2008. GCE '08, pp.1-10, 12-16 Nov. 2008.




Authorized licensed use limited to: Mukesh Patel School of Technology & Engineering. Downloaded on July 11,2023 at 05:39:03 UTC from IEEE Xplore. Restrictions apply.